MPRI 2.30: Proofs of Security Protocols

3. Security Proofs, Authentication

Adrien Koutsos, Inria Paris

2024/2025

Security Proof

Protocol Branching

We consider a more useful version of PA in which S checks whether it is talking to I or not.

The PA Protocol, v2

$$1 : \mathsf{I} \to \mathsf{S} : \nu\, \mathsf{n_I}. \qquad \mathbf{out}(\mathtt{I}, \{\langle \mathsf{pk_I}\,, \mathsf{n_I}\rangle\}_{\mathsf{pk_S}})$$

$$2 : \mathsf{S} \to \mathsf{I} : \nu\, \mathsf{n_S}. \mathbf{in}(\mathtt{S}, x).\ \mathbf{out}(\mathtt{S}, \text{if } \pi_1(d) = \mathsf{pk_I} \text{ then } \{\langle \pi_2(d)\,, \mathsf{n_S}\rangle\}_{\mathsf{pk_I}} \text{ else } \{0\}_{\mathsf{pk_I}})$$

where $d \equiv \mathsf{dec}(\mathsf{x}, \mathsf{sk_S})$.

The encryption of 0 in the else branch is here to hide to the adversary which branch was taken.

Private Authentication: Anonymity

Lets now try to prove that PA v2 provides **anonymity**:

- I_X is the initiator with identity X;
- S_X is the server, accepting messages from X;

The adversary must not be able to distinguish $I_A \mid S_A$ from $I_C \mid S_A$.

$$I_X : \nu\, r.\ \nu\, n_I. \qquad \mathbf{out}(\mathtt{I}, \{\langle pk_X , n_I\rangle\}^{r}_{pk_S})$$

$$S_X : \nu\, r_0. \nu\, n_S. \mathbf{in}(\mathtt{S}, x). \ \mathbf{out}(\mathtt{S}, \text{if } \pi_1(d_X) = pk_X \text{ then } \{\langle \pi_2(d_X) , n_S\rangle\}^{r_0}_{pk_X} \text{ else } \{0\}^{r_0}_{pk_X})$$

We assume the encryption is IND-CCA$_1$ and KP-CCA$_1$.

Private Authentication: Anonymity

As we saw, an encryption **does not hide the length** of the plain-text. Hence, since $\mathsf{len}(\langle \mathsf{n_I}, \mathsf{n_S} \rangle) \neq \mathsf{len}(0)$, there is an attack:

$$\not\models \{\langle \mathsf{n_I}, \mathsf{n_S} \rangle\}^{\mathsf{r_0}}_{\mathsf{pk_A}} \sim \{0\}^{\mathsf{r_0}}_{\mathsf{pk_C}}$$

even if the encryption is IND-CCA$_1$ and KP-CCA$_1$.

Private Authentication: Anonymity

We fix the protocol by:

- adding a **length check**;
- using a **decoy** message of the correct length.

The PA Protocol, v3

$$\mathsf{I_X} : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}. \qquad \mathbf{out}(\mathtt{I}, \{\langle \mathsf{pk_X}, \mathsf{n_I} \rangle\}^{\mathsf{r}}_{\mathsf{pk_S}})$$

$$\begin{aligned}\mathsf{S_X} : \nu\, \mathsf{r_0}. \nu\, \mathsf{n_S}. \mathbf{in}(\mathtt{S}, x). \ \mathbf{out}(\mathtt{S}, &\text{if } \pi_1(d_{\mathsf{X}}) = \mathsf{pk_X} \wedge \mathsf{len}(\pi_2(d_{\mathsf{X}})) = \mathsf{len}(\mathsf{n_S})) \\ &\text{then } \{\langle \pi_2(d_{\mathsf{X}}), \mathsf{n_S} \rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}} \\ &\text{else } \ \{\langle \mathsf{n_S}, \mathsf{n_S} \rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}\end{aligned}$$

Private Authentication: Anonymity

$$\mathsf{I_X} : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}. \quad \mathbf{out}(\mathtt{I}, \{\langle \mathsf{pk_X}, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}})$$

$$\mathsf{S_X} : \nu\, \mathsf{r_0}. \nu\, \mathsf{n_S}. \mathbf{in}(\mathtt{S}, x). \mathbf{out}(\mathtt{S}, \text{if } \pi_1(d_\mathsf{X}) = \mathsf{pk_X} \wedge \mathsf{len}(\pi_2(d_\mathsf{X})) = \mathsf{len}(\mathsf{n_S}))$$

$$\text{then } \{\langle \pi_2(d_\mathsf{X}), \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}$$

$$\text{else } \{\langle \mathsf{n_S}, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}$$

To prove $\mathsf{I_A} \mid \mathsf{S_A} \approx \mathsf{I_C} \mid \mathsf{S_A}$, we have several **traces**:

$\mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{I}), \mathbf{out}(\mathtt{S})$	$\mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{S}), \mathbf{out}(\mathtt{I})$	$\mathbf{out}(\mathtt{I}), \mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{S})$
$\mathbf{out}(\mathtt{I}), \mathbf{out}(\mathtt{S}), \mathbf{in}(\mathtt{S})$	$\mathbf{out}(\mathtt{S}), \mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{I})$	$\mathbf{out}(\mathtt{S}), \mathbf{out}(\mathtt{S}), \mathbf{in}(\mathtt{S})$

Private Authentication: Anonymity

$$\mathsf{I_X} : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}.\quad \mathbf{out}(\mathtt{I}, \{\langle \mathsf{pk_X}, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}})$$

$$\mathsf{S_X} : \nu\, \mathsf{r_0}.\, \nu\, \mathsf{n_S}.\, \mathbf{in}(\mathtt{S}, x).\ \mathbf{out}(\mathtt{S}, \text{if } \pi_1(d_X) = \mathsf{pk_X} \wedge \mathsf{len}(\pi_2(d_X)) = \mathsf{len}(\mathsf{n_S}))$$

$$\text{then } \{\langle \pi_2(d_X), \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}$$

$$\text{else } \{\langle \mathsf{n_S}, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}$$

To prove $\mathsf{I_A} \mid \mathsf{S_A} \approx \mathsf{I_C} \mid \mathsf{S_A}$, we have several **traces**:

$\mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{I}), \mathbf{out}(\mathtt{S})$	$\mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{S}), \mathbf{out}(\mathtt{I})$	$\mathbf{out}(\mathtt{I}), \mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{S})$
$\mathbf{out}(\mathtt{I}), \mathbf{out}(\mathtt{S}), \mathbf{in}(\mathtt{S})$	$\mathbf{out}(\mathtt{S}), \mathbf{in}(\mathtt{S}), \mathbf{out}(\mathtt{I})$	$\mathbf{out}(\mathtt{S}), \mathbf{out}(\mathtt{S}), \mathbf{in}(\mathtt{S})$

But there is a **more general trace**: its security implies the security of the other traces.

See **partial order reduction** (POR) techniques [1].

Private Authentication: Anonymity

We must prove that:

$$\mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \sim \mathsf{out}_1^{\mathsf{C}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{C}}]$$

where:

$$\begin{aligned}
\mathsf{out}_1^{\mathsf{X}} &\equiv \{\langle \mathsf{pk}_{\mathsf{X}} \, , \, \mathsf{n}_{\mathsf{I}} \rangle\}_{\mathsf{pk}_{\mathsf{S}}}^{\mathsf{r}} \\
\mathsf{out}_2^{\mathsf{X},\mathsf{Y}}[\mathsf{M}] &\equiv \text{if } \pi_1(d[\mathsf{M}]) = \mathsf{pk}_{\mathsf{X}} \wedge \mathsf{len}(\pi_2(d[\mathsf{M}])) = \mathsf{len}(\mathsf{n}_{\mathsf{S}}) \\
&\qquad \text{then } \{\langle \pi_2(d[\mathsf{M}]) \, , \, \mathsf{n}_{\mathsf{S}} \rangle\}_{\mathsf{pk}_{\mathsf{Y}}}^{\mathsf{r}_0} \\
&\qquad \text{else } \ \{\langle \mathsf{n}_{\mathsf{S}} \, , \, \mathsf{n}_{\mathsf{S}} \rangle\}_{\mathsf{pk}_{\mathsf{Y}}}^{\mathsf{r}_0} \\
d[\mathsf{M}] &\equiv \mathsf{dec}(\mathbf{att}_0([\mathsf{M}]), \mathsf{sk}_{\mathsf{S}})
\end{aligned}$$

Private Authentication: Anonymity

First, we push the branching under the encryption:

$$\frac{\mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \sim \mathsf{out}_1^{C}, \underline{\mathsf{out}_2}^{A,A}[\mathsf{out}_1^{C}] \quad \overline{\mathsf{out}_2^{A,A}[\mathsf{out}_1^{C}] = \underline{\mathsf{out}_2}^{A,A}[\mathsf{out}_1^{C}]}}{\mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \sim \mathsf{out}_1^{C}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{C}]}\ \mathrm{R}$$

where:

$$\underline{\mathsf{out}_2}^{X,Y}[M] \ \equiv \ \left\{ \begin{array}{l} \text{if } \pi_1(d[M]) = \mathsf{pk}_X \wedge \mathsf{len}(\pi_2(d[M])) = \mathsf{len}(n_S) \\ \ \text{then } \langle \pi_2(d[M])\,,\, n_S \rangle \\ \ \text{else } \ \langle n_S\,,\, n_S \rangle \end{array} \right\}_{\mathsf{pk}_Y}^{r_0}$$

We let $m_X[M]$ be the content of the encryption above.

Private Authentication: Anonymity

Then, we use KP-CCA$_1$ to change the encryption key:

$$\dfrac{\begin{array}{ll} & \mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \\ \sim & \mathsf{out}_1^{C}, \underline{\mathsf{out}_2}^{A,C}[\mathsf{out}_1^{C}] \end{array} \qquad \begin{array}{ll} & \dfrac{}{\mathsf{out}_1^{C}, \underline{\mathsf{out}_2}^{A,C}[\mathsf{out}_1^{C}]}\ \textsc{KP-CCA}_1 \\ \sim & \mathsf{out}_1^{C}, \underline{\mathsf{out}_2}^{A,A}[\mathsf{out}_1^{C}] \end{array}}{\mathsf{out}_1^{A}, \mathsf{out}_2^{A,A}[\mathsf{out}_1^{A}] \sim \mathsf{out}_1^{C}, \underline{\mathsf{out}_2}^{A,A}[\mathsf{out}_1^{C}]}\ \textsc{Trans}$$

since:

- the encryption randomness r_0 is correctly used;
- the key randomness n_A and n_B appear only in $\mathsf{pk}(\cdot)$ and $\mathsf{dec}(\_, \mathsf{sk}(\cdot))$ positions.

Private Authentication: Anonymity

Then, we use IND-CCA$_1$ to change the encryption content:

$$\dfrac{\begin{array}{ll} & \mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \\ \sim & \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{C},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}] \end{array} \qquad \begin{array}{ll} & \dfrac{\left[\mathsf{len}(m_{\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]) = \mathsf{len}(m_{\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}])\right]}{\mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{C},\mathsf{C}}[\mathsf{out}_1^{\mathsf{A}}]}\ \textsc{ind-cca}_1 \\ \sim & \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}] \end{array}}{\mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \sim \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{A},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]}\ \textsc{Trans}$$

since:

- the encryption randomness r_0 is correctly used;
- the key randomness n_C appear only in $\mathsf{pk}(\cdot)$ and $\mathsf{dec}(\_, \mathsf{sk}(\cdot))$ positions.

Private Authentication: Anonymity

Recall that:

$$m_X[M] \equiv \text{if } \pi_1(d[M]) = \mathsf{pk}_X \wedge \mathsf{len}(\pi_2(d[M])) = \mathsf{len}(n_S) \\ \text{then } \langle \pi_2(d[M]) , n_S \rangle \\ \text{else } \langle n_S , n_S \rangle$$

Then:

$$\frac{\mathcal{A}_{th} \vdash_{\text{GEN}} \mathsf{len}(m_C[\mathsf{out}_1^C]) = \mathsf{len}(m_A[\mathsf{out}_1^A])}{\left[\mathsf{len}(m_C[\mathsf{out}_1^C]) = \mathsf{len}(m_A[\mathsf{out}_1^A])\right]} \text{ GEN}$$

if $\mathcal{A}_{th}$ contains the axiom[1]:

$$\forall x, y. \mathsf{len}(\langle x , y \rangle) = c_{\langle \_ , \_ \rangle}(\mathsf{len}(x), \mathsf{len}(y))$$

where $c_{\langle \_ , \_ \rangle}(\cdot, \cdot)$ is left unspecified.

[1]This axiom must be satisfied by the protocol implementation for the security proof to apply.

Private Authentication: Anonymity

Then, we α-rename the key randomness n_{C}, rewrite back the encryption, and conclude.

$$\frac{}{\mathsf{out}_1^{\mathsf{A}}, \mathsf{out}_2^{\mathsf{A},\mathsf{A}}[\mathsf{out}_1^{\mathsf{A}}] \sim \mathsf{out}_1^{\mathsf{C}}, \underline{\mathsf{out}_2}^{\mathsf{C},\mathsf{C}}[\mathsf{out}_1^{\mathsf{C}}]}\ \alpha\text{-}\textsc{equ} + \mathrm{R} + \textsc{Refl}$$

Privacy

Privacy

We proved **anonymity** of the Private Authentication protocol, which we defined as:

$$I_A \mid S_A \approx I_C \mid S_A$$

But does this really guarantees that this protocol protects the privacy of its users?
$\Rightarrow$ **No, because of linkability attacks**

Linkability Attacks

Consider the following authentication protocol, called KCL, between a reader R and a tag $\mathsf{T_X}$ with identity X:

$$\begin{aligned} \mathsf{R} \quad &: \nu\, \mathsf{n_R}. \qquad\qquad\quad \mathbf{out}(\mathtt{R}, \mathsf{n_R}) \\ \mathsf{T_X} &: \nu\, \mathsf{n_T}. \mathbf{in}(\mathtt{T}, \mathtt{x}). \mathbf{out}(\mathtt{T}, \langle \mathsf{X} \oplus \mathsf{n_T}, \mathsf{n_T} \oplus \mathsf{H}(\mathtt{x}, \mathsf{k_X}) \rangle) \end{aligned}$$

Assuming H is a PRF (Pseudo-Random Function), and $\oplus$ is the exclusive-or, we can prove that KCL provides **anonymity**.

$$\mathsf{T_A} \mid \mathsf{R} \approx \mathsf{T_B} \mid \mathsf{R}$$

Linkability Attacks

But there are **privacy attacks** against KCL, using two sessions:

$$
\begin{array}{ll|ll}
1: E \rightarrow T_A : n_R & & E \rightarrow T_A : n_R & \\
2: T_A \rightarrow E : \langle A \oplus n_T \,,\, n_T \oplus H(n_R, k_A)\rangle & & T_A \rightarrow E : \langle A \oplus n_T \,,\, n_T \oplus H(n_R, k_A)\rangle & \\
 & & & \\
3: E \rightarrow T_A : n_R & & E \rightarrow T_B : n_R & \\
4: T_A \rightarrow E : \langle A \oplus n'_T \,,\, n'_T \oplus H(n_R, k_A)\rangle & & T_B \rightarrow E : \langle B \oplus n'_T \,,\, n'_T \oplus H(n_R, k_B)\rangle &
\end{array}
$$

Let t_2 and t_4 be the outputs of T. Then, on the left scenario:

$$
\begin{aligned}
\pi_2(t_2) \oplus \pi_2(t_4) &= \big(n_T \oplus H(n_R, k_A)\big) \oplus \big(n'_T \oplus H(n_R, k_A)\big) \\
&= n_T \oplus n'_T \\
&= \pi_1(t_2) \oplus \pi_1(t_4)
\end{aligned}
$$

The same equality check will almost never hold on the right, under reasonable assumption on H.

Linkability Attacks

We just saw an **attack** against:

$$(\mathsf{T_A} \mid \mathsf{R}) \mid (\mathsf{T_A} \mid \mathsf{R}) \approx (\mathsf{T_A} \mid \mathsf{R}) \mid (\mathsf{T_B} \mid \mathsf{R})$$

Unlinkability

To prevent such attacks, we need to prove a stronger property, called **unlinkability**. It requires to prove the **equivalence** between:

- a **real-world**, where each agent can run **many sessions**:

$$\nu\,\vec{k}_0, \ldots, \vec{k}_N.\ !_{\mathrm{id} \leq N}\ !_{\mathtt{sid} \leq M}\ P(\vec{k}_{\mathrm{id}})$$

- and an **ideal-world**, where each agent run at most a **single session**:

$$\nu\,\vec{k}_{0,0}, \ldots, \vec{k}_{N,M}.\ !_{\mathrm{id} \leq N}\ !_{\mathtt{sid} \leq M}\ P(\vec{k}_{\mathrm{id},\mathtt{sid}})$$

Notation: $!_{x \leq N}\ \mathsf{P}(x)$ is the replication of the process P, and is syntactic sugar for $\mathsf{P}(0), \ldots, \mathsf{P}(N)$.

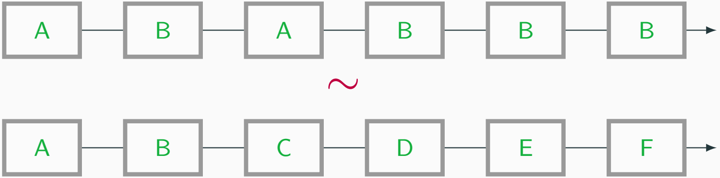
Remark

The processes above are parameterized by $N, M \in \mathbb{N}$. Unlinkability holds if the equivalence holds for any N, M.

For the sack of simplicity, we omit channel names.

Unlinkability

Example An unlinkability scenario.



Unlinkability: Intuition

In the **ideal-world**, relations between sessions **cannot leak** any **information** on identities.
$\Rightarrow$ hence **no link** can be **efficiently found** in the **real word**.

Unlinkability: Adding Servers

Our definition of **unlinkability** did not account for the **server**.

User-specific server, accepting a single identity.
The processes $P(\vec{s}, \vec{\mathsf{k}}_{\mathsf{U}})$ and $S(\vec{\mathsf{k}}_{\mathsf{S}}, \vec{\mathsf{k}}_{\mathsf{U}})$ are parameterized by:

- **global** key material $\vec{s}$;
- and **user-specific** key material $\vec{\mathsf{k}}_{\mathsf{U}}$.

Then, we require that:

$$\begin{aligned} & \nu\, \vec{s}.\, \nu\, \vec{\mathsf{k}}_0, \ldots, \vec{\mathsf{k}}_N. \quad !_{\mathsf{id} \leq N}\ !_{\mathtt{sid} \leq M}\ \big(P(\vec{s}, \vec{\mathsf{k}}_{\mathsf{id}}) \quad \mid S(\vec{s}, \vec{\mathsf{k}}_{\mathsf{id}})\big) \\ \approx\ & \nu\, \vec{s}.\, \nu\, \vec{\mathsf{k}}_{0,0}, \ldots, \vec{\mathsf{k}}_{N,M}.\ !_{\mathsf{id} \leq N}\ !_{\mathtt{sid} \leq M}\ \big(P(\vec{s}, \vec{\mathsf{k}}_{\mathsf{id},\mathtt{sid}}) \mid S(\vec{s}, \vec{\mathsf{k}}_{\mathsf{id},\mathtt{sid}})\big) \end{aligned}$$

Unlinkability: Adding Servers

Generic server, accepting all identities.

No changes for the user process $P(\vec{s}, \vec{\mathsf{k}}_{\mathsf{U}})$.

The server $S(\vec{s}, \vec{\mathsf{k}}_0, \ldots, \vec{\mathsf{k}}_M)$ is parameterized by:

- some **global** key material $\vec{s}$;
- **all users** key material $\vec{\mathsf{k}}_0, \ldots, \vec{\mathsf{k}}_M$.

Then we require that:

$$
\begin{aligned}
\nu\, \vec{s}.\, \nu\, \vec{\mathsf{k}}_0, \ldots, \vec{\mathsf{k}}_N. \quad & \left(!_{\mathsf{id} \le N}\ !_{\mathtt{sid} \le M}\ P(\vec{s}, \vec{\mathsf{k}}_{\mathsf{id}})\right) \ \mid \\
& \left(!_{\le L}\ S(\vec{s}, \vec{\mathsf{k}}_0, \ldots, \vec{\mathsf{k}}_N)\right) \\
\approx \nu\, \vec{s}.\, \nu\, \vec{\mathsf{k}}_{0,0}, \ldots, \vec{\mathsf{k}}_{N,M}. \quad & \left(!_{\mathsf{id} \le N}\ !_{\mathtt{sid} \le M}\ P(\vec{s}, \vec{\mathsf{k}}_{\mathsf{id},\mathtt{sid}})\right) \ \mid \\
& \left(!_{\le L}\ S(\vec{s}, \vec{\mathsf{k}}_{0,0}, \ldots, \vec{\mathsf{k}}_{N,M})\right)
\end{aligned}
$$

Unlinkability: Remark

Note that **user-specific unlinkability** is a very strong property that does not often hold.

Example

Assume S **leaks whether it succeeded** or not. This models the fact that the adversary can **distinguish success from failure**:

- e.g. because a door opens, which can be observed;
- or because success is followed by further communication, while failure is followed by a new authentication attempt.

Then the following unlinkability scenario **does not hold**:

$$\Big(\underline{P(\vec{k})} \mid S(\vec{k})\Big) \mid \Big(P(\vec{k}) \mid \underline{S(\vec{k})}\Big) \not\approx \Big(\underline{P(\vec{k_0})} \mid S(\vec{k_0})\Big) \mid \Big(P(\vec{k_1}) \mid \underline{S(\vec{k_1})}\Big)$$

✓ ✗

Private Authentication: Unlinkability

Private Authentication

We parameterize the initiator and server in **PA** by the key material:

$$\mathsf{I}(\mathsf{k_S}, \mathsf{k_X}) : \nu\, \mathsf{r}.\ \nu\, \mathsf{n_I}.\ \ \mathbf{out}(\mathtt{I}, \{\langle \mathsf{pk_X}, \mathsf{n_I}\rangle\}^{\mathsf{r}}_{\mathsf{pk_S}})$$

$$\begin{aligned}\mathsf{S}(\mathsf{k_S}, \mathsf{k_X}) : \nu\, \mathsf{r_0}.\, \nu\, \mathsf{n_S}.\, \mathbf{in}(\mathtt{S}, x).\ \mathbf{out}(\mathtt{S}, &\text{if } \pi_1(d) = \mathsf{pk_X} \wedge \mathsf{len}(\pi_2(d)) = \mathsf{len}(\mathsf{n_S})) \\ &\text{then } \{\langle \pi_2(d), \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}} \\ &\text{else } \{\langle \mathsf{n_S}, \mathsf{n_S}\rangle\}^{\mathsf{r_0}}_{\mathsf{pk_X}}\end{aligned}$$

where $\mathsf{sk_X} \equiv \mathsf{sk}(\mathsf{k_X})$, $\mathsf{pk_X} \equiv \mathsf{pk}(\mathsf{k_X})$ and $d \equiv \mathsf{dec}(x, \mathsf{sk_S})$.

Private Authentication: Unlinkability

Theorem

Private Authentication, v3 satisfies the **unlinkability** property (with user-specific server). I.e., for all $N, M \in \mathbb{N}$:

$$\begin{aligned} & \nu\, \mathsf{k}_{\mathsf{S}}.\, \nu\, \mathsf{k}_0, \ldots, \mathsf{k}_N. \quad !_{\mathtt{id} \leq N}\, !_{\mathtt{sid} \leq M} \left(I(\mathsf{k}_{\mathsf{S}}, \mathsf{k}_{\mathtt{id}}) \mid S(\mathsf{k}_{\mathsf{S}}, \mathsf{k}_{\mathtt{id}}) \right) \\ \approx\ & \nu\, \mathsf{k}_{\mathsf{S}}.\, \nu\, \mathsf{k}_{0,0}, \ldots, \mathsf{k}_{N,M}.\, !_{\mathtt{id} \leq N}\, !_{\mathtt{sid} \leq M} \left(I(\mathsf{k}_{\mathsf{S}}, \mathsf{k}_{\mathtt{id},\mathtt{sid}}) \mid S(\mathsf{k}_{\mathsf{S}}, \mathsf{k}_{\mathtt{id},\mathtt{sid}}) \right) \end{aligned}$$

Proof sketch

For all N, M, for all trace of observables $\mathtt{tr}$, we show that:

$$\models \text{s-exec}(\mathsf{P}_{\mathcal{L}}, \mathtt{tr}) \sim \text{s-exec}(\mathsf{P}_{\mathcal{R}}, \mathtt{tr})$$

by induction over $\mathtt{tr}$, where $\mathsf{P}_{\mathcal{L}}$ and $\mathsf{P}_{\mathcal{R}}$ are, resp., the left and right protocols in the theorem above.

Authentication Protocols

Authentication Protocol

We now focus on another class of security properties: **correspondance properties** (e.g. **authentication**)

These are properties on a **single** protocol, often expressed as a **temporal** property on **events** of the protocol. E.g.

> *If **Alice** accepts **Bob** at time τ then **Bob** must have initiated a session with **Alice** at time $\tau' < \tau$.*

To formalize the **cryptographic arguments** proving such properties, we will design a specialized **framework** and **proof system**.

The Hash-Lock Protocol

Let $\mathcal{I}$ be a finite set of identities.

Hash-Lock

$$\mathsf{T}(\mathsf{A},\mathtt{i}) : \nu\, \mathsf{n}_{\mathsf{A},\mathtt{i}}.\ \mathbf{in}(\mathtt{A}_\mathtt{i}, \mathsf{x}).\ \mathbf{out}(\mathtt{A}_\mathtt{i}, \langle \mathsf{n}_{\mathsf{A},\mathtt{i}}, \mathsf{H}(\langle \mathsf{x}, \mathsf{n}_{\mathsf{A},\mathtt{i}}\rangle, \mathsf{k}_\mathsf{A})\rangle)$$

$$\begin{aligned}\mathsf{R}(\mathtt{j}) : \nu\, \mathsf{n}_{\mathsf{R},\mathtt{j}}.\ & \mathbf{in}(\mathtt{R}^1_\mathtt{j}, \_).\ \mathbf{out}(\mathtt{R}^1_\mathtt{j}, \mathsf{n}_{\mathsf{R},\mathtt{j}}).\\ & \mathbf{in}(\mathtt{R}^2_\mathtt{j}, \mathsf{y}).\\ & \mathbf{out}(\mathtt{R}^2_\mathtt{j}, \text{if } \bigvee_{\mathsf{A}\in\mathcal{I}} \pi_2(\mathsf{y}) = \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}}, \pi_1(\mathsf{y})\rangle, \mathsf{k}_\mathsf{A}))\\ & \qquad \text{then ok}\\ & \qquad \text{else ko}\end{aligned}$$

We consider N sessions of each tag, and M sessions of the reader:

$$\nu\,(\mathsf{k}_\mathsf{A})_{\mathsf{A}\in\mathcal{I}}.\ \left(!_{\mathsf{A}\in\mathcal{I}}\ !_{\mathtt{i}<N}\ \mathsf{T}(\mathsf{A},\mathtt{i})\right)\ \mid\ \left(!_{\mathtt{j}<M}\ \mathsf{R}(\mathtt{j})\right)$$

Remark: we abuse notations and write $\mathtt{R}^\mathtt{i}_\mathtt{j}$ to denote the *i*-th usage of channel $\mathtt{R}_\mathtt{j}$ in a process.

Authentication

Definition(informal)

If the j-th session of R accepts believing it talked to tag A, then:

- there exists a session i of tag A **properly interleaved** with the j-th session of R;
- **messages** have been **properly forwarded** between the i-th session of tag A and the j-th session of R.

💡 *The second condition is often relaxed to require only a partial correspondence between messages.*

Next slides: a framework to express such temporal properties.

Notations

- we let $\leq$ be the **prefix relation** over observable traces:

$$\mathtt{tr}_0 \leq \mathtt{tr}_1 \quad \text{iff.} \quad \exists \mathtt{tr}'.\, \mathtt{tr}_1 = \mathtt{tr}_0; \mathtt{tr}'$$

- $\mathtt{tr} : \mathrm{c}$ states that $\mathtt{tr}$ **ends with an output** on c:

$$\mathtt{tr} : \mathrm{c} \quad \text{iff.} \quad \exists \mathtt{tr}'.\, \mathtt{tr} = \mathtt{tr}'; \mathbf{out}(\mathrm{c})$$

- $\mathtt{tr} : \mathrm{c}^n$ means that $\mathtt{tr} : \mathrm{c}$ and $\mathtt{tr}$ **contains n outputs on** c:

$$\mathtt{tr} : \mathrm{c}^n \quad \text{iff.} \quad \begin{cases} \text{true} & \text{if } n = 0 \\ \exists \mathtt{tr}_0, \mathtt{tr}_1.\, \mathtt{tr} = \mathtt{tr}_0, \mathtt{tr}_1 \wedge & \text{otherwise} \\ \qquad\qquad \mathtt{tr}_0 : \mathrm{c}^{n-1} \wedge & \\ \qquad\qquad \mathtt{tr}_1 : \mathrm{c}^1 & \end{cases}$$

Notation: $\mathtt{tr} : \mathrm{c}^n \leq \mathtt{tr}'$ means $\mathtt{tr} : \mathrm{c}^n \wedge \mathtt{tr} \leq \mathtt{tr}'$.

POR Result (Assumed)

We let $\mathcal{T}_{\mathsf{io}}$ be the set of observable traces where all outputs are always **directly preceded** by an input on the same channel, i.e.:

$$\mathtt{tr} \in \mathcal{T}_{\mathsf{io}} \quad \text{iff.} \quad \forall \mathtt{tr}' : \mathtt{c} \leq \mathtt{tr}.\, \exists \mathtt{tr}''.\, \mathtt{tr}' = \mathtt{tr}''; \mathbf{in}(\mathtt{c}); \mathbf{out}(\mathtt{c})$$

Assumption: POR

We admit that to analyze the Hash-Lock protocol, it is sufficient to consider only observables traces in $\mathcal{T}_{\mathsf{io}}$.

Authentication of the Hash-Lock Protocol

For any $\mathtt{tr} : \mathtt{R}_\mathtt{j}^2 \in \mathcal{T}_{\mathsf{io}}$, we let $\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}$ be a term (defined later) stating that the reader accepts the tag A at the end of the trace $\mathtt{tr}$.

Authentication of the Hash-Lock Protocol

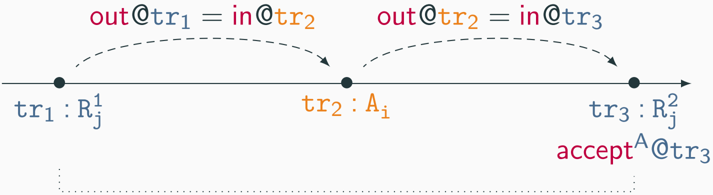
Informally, Hash-Lock provides **authentication** if for all $\mathtt{tr} \in \mathcal{T}_{io}$, $\mathtt{tr}_1 : \mathtt{R}_j^1$ and $\mathtt{tr}_3 : \mathtt{R}_j^2$ such that:

$$\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr} \qquad \text{and} \qquad \mathsf{accept}^{A}@\mathtt{tr}_3$$

there must exists $\mathtt{tr}_2 : \mathtt{A}_i$ such that $\mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3$ and:

$$\mathsf{out}@\mathtt{tr}_1 = \mathsf{in}@\mathtt{tr}_2 \wedge \mathsf{out}@\mathtt{tr}_2 = \mathsf{in}@\mathtt{tr}_3$$

Graphically:



Authentication of the Hash-Lock Protocol

What do we lack to formalize and prove the **authentication** of the **Hash-Lock** protocol?

- define the (generic) **terms representing** the output, input and acceptance, which we need to state the property;
- have a set of rules for $[\,\cdot\,]$ that can capture the security proof.

Authentication Protocols

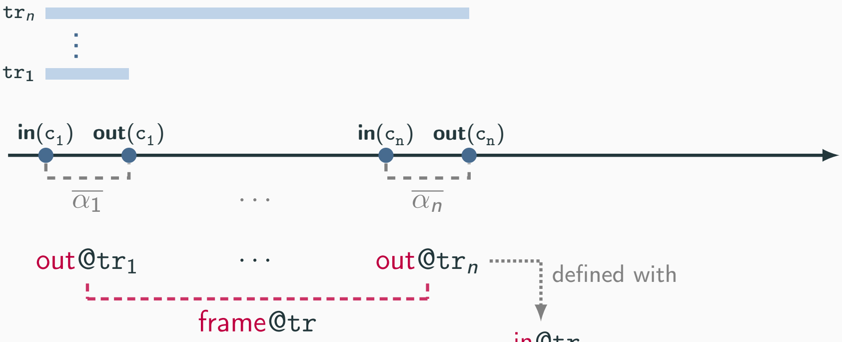
Macro Terms

Notations: Predecessor

For any **observable trace** $\texttt{tr}$ and **observable** α, we let:

$$\mathsf{pred}(\texttt{tr}; \alpha) \stackrel{\mathsf{def}}{=} \texttt{tr}$$

Macro Terms: Graphical Representation



$$\text{in@}\mathtt{tr}_n$$
$$= \mathbf{att}_{n-1}(\text{frame@}\mathtt{tr}_{n-1})$$
$$= \mathbf{att}_{n-1}(\text{out@}\mathtt{tr}_1, \ldots, \text{out@}\mathtt{tr}_{n-1})$$

Macro Terms

We now define some **generic** terms and sequences of terms by **induction** of the observable trace $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$.

Let $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$ with n inputs. If $\text{s-exec}(\mathsf{P}, \mathtt{tr}) = t_1, \ldots, t_n$ then we let:

$$\mathsf{out}_{\mathsf{P}}@\mathtt{tr} \overset{\text{def}}{=} \begin{cases} t_n & \text{if } \exists \mathtt{c}.\ \mathtt{tr} \neq \epsilon \\ \text{empty} & \text{otherwise} \end{cases}$$

$$\mathsf{frame}_{\mathsf{P}}@\mathtt{tr} \overset{\text{def}}{=} \begin{cases} \mathsf{frame}_{\mathsf{P}}@\mathsf{pred}(\mathtt{tr}), \mathsf{out}_{\mathsf{P}}@\mathtt{tr} & \text{if } \mathtt{tr} \neq \epsilon \\ \epsilon & \text{if } \mathtt{tr} = \epsilon \end{cases}$$

$$\mathsf{in}_{\mathsf{P}}@(\mathtt{tr}) \overset{\text{def}}{=} \begin{cases} \mathbf{att}_{n-1}(\mathsf{frame}_{\mathsf{P}}@\mathsf{pred}(\mathtt{tr})) & \text{if } \mathtt{tr} \neq \epsilon \\ \text{empty} & \text{if } \mathtt{tr} = \epsilon \end{cases}$$

Remark: we omit P when it is clear from context.

- *The restriction to traces in $\mathcal{T}_{io}$ simplifies the definition of $\mathsf{in}_P@\mathtt{tr}$.*
- *$\mathsf{frame}_P@\mathtt{tr}$ is an alternative name for s-exec(P, $\mathtt{tr}$).*

Hash-Lock: Accept

Hash-Lock

$$
\begin{aligned}
\mathsf{T}(\mathsf{A}, \mathtt{i}) &: \nu\, \mathsf{n}_{\mathsf{A},\mathtt{i}}.\ \mathbf{in}(\mathtt{A_i}, \mathsf{x}).\ \mathbf{out}(\mathtt{A_i}, \langle \mathsf{n}_{\mathsf{A},\mathtt{i}}\,,\ \mathsf{H}(\langle \mathsf{x}\,,\ \mathsf{n}_{\mathsf{A},\mathtt{i}} \rangle, \mathsf{k_A}) \rangle) \\
\mathsf{R}(\mathtt{j}) &: \nu\, \mathsf{n}_{\mathsf{R},\mathtt{j}}.\ \mathbf{in}(\mathtt{R}^1_\mathtt{j}, \_).\ \mathbf{out}(\mathtt{R}^1_\mathtt{j}, \mathsf{n}_{\mathsf{R},\mathtt{j}}). \\
&\qquad \mathbf{in}(\mathtt{R}^2_\mathtt{j}, \mathsf{y}). \\
&\qquad \mathbf{out}(\mathtt{R}^2_\mathtt{j}, \text{if } \bigvee_{\mathsf{A}\in\mathcal{I}} \pi_2(\mathsf{y}) = \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\ \pi_1(\mathsf{y})\rangle, \mathsf{k_A})) \\
&\qquad\qquad \text{then ok} \\
&\qquad\qquad \text{else ko}
\end{aligned}
$$

To be able to state some **authentication** property of Hash-Lock, we need an additional macro. For all $\mathtt{tr} : \mathtt{R}^2_\mathtt{j} \in \mathcal{T}_{\mathsf{io}}$, we let:

$$
\mathsf{accept}^{\mathsf{A}}@\mathtt{tr} \stackrel{\text{def}}{=} \pi_2(\mathsf{in}@\mathtt{tr}) = \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\ \pi_1(\mathsf{in}@\mathtt{tr})\rangle, \mathsf{k_A})
$$

We made sure that all names in the protocol are unique, so that they don't have to be renamed before the symbolic execution.

Authentication: Hash-Lock

The following formulas encode the fact that the **Hash-Lock** protocol provides **authentication**:

$$\forall \mathsf{A} \in \mathcal{I}.\ \forall \mathtt{tr} \in \mathcal{T}_{\mathsf{io}}.\ \forall \mathtt{tr}_1 : \mathtt{R}^1_{\mathtt{j}}, \mathtt{tr}_3 : \mathtt{R}^2_{\mathtt{j}} \text{ s.t. } \mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr},$$

$$\left[\mathsf{accept}^{\mathsf{A}} @ \mathtt{tr}_3 \rightarrow \bigvee_{\substack{\mathtt{tr}_2 : \mathtt{A}_\mathtt{i} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}} \begin{array}{l} \mathsf{out}@\mathtt{tr}_1 = \mathsf{in}@\mathtt{tr}_2 \wedge \\ \mathsf{out}@\mathtt{tr}_2 = \mathsf{in}@\mathtt{tr}_3 \end{array} \right]$$

This kind of one-sided properties are called **correspondance** properties. Proving their validity will require **additional rules**, to allow for **propositional reasoning**.

Authentication Protocols

Local Proof System

We define a **judgment** dedicated to **correspondance properties**.

Definition

A **local judgement** $\Gamma \vdash t$ comprises a sequence of boolean terms $\Gamma = \phi_1, \ldots, \phi_n$ and a boolean term ϕ.

$\Gamma \vdash \phi$ is **valid** if and only if the following formula is valid:

$$[\phi_1 \to \cdots \to \phi_n \to \phi]$$

Boolean Connectives in Local Judgements

Careful not to confuse the boolean connectives at the **local** and **equivalence** levels!

Exercise

Determine which directions are correct.

$$[\phi \wedge \psi] \quad \overset{?}{\Leftrightarrow} \quad [\phi] \mathbin{\tilde{\wedge}} [\psi]$$

$$[\phi \vee \psi] \quad \overset{?}{\Leftrightarrow} \quad [\phi] \mathbin{\tilde{\vee}} [\psi]$$

$$[\phi \rightarrow \psi] \quad \overset{?}{\Leftrightarrow} \quad [\phi] \mathbin{\tilde{\rightarrow}} [\psi]$$

Boolean Connectives in Local Judgements

Careful not to confuse the boolean connectives at the **local** and **equivalence** levels!

Exercise

Determine which directions are correct.

$$[\phi \wedge \psi] \quad \Leftrightarrow \quad [\phi] \mathbin{\tilde{\wedge}} [\psi]$$

$$[\phi \vee \psi] \quad \Leftarrow \quad [\phi] \mathbin{\tilde{\vee}} [\psi]$$

$$[\phi \rightarrow \psi] \quad \Rightarrow \quad [\phi] \mathbin{\tilde{\rightarrow}} [\psi]$$

The second relation works both ways when ϕ or ψ is a **constant** formula.

Local Proof System

Our **local judgement** can be trivially equipped with a **sequent calculus** that behaves as a standard FO sequent calculus.

$$\frac{}{\Gamma, \phi \vdash \phi} \qquad \frac{\Gamma \vdash \psi \qquad \Gamma, \psi \vdash \phi}{\Gamma \vdash \phi}$$

$$\frac{\Gamma \vdash \psi \qquad \Gamma \vdash \phi}{\Gamma \vdash \psi \wedge \phi} \qquad \frac{\Gamma, \psi, \phi \vdash \theta}{\Gamma, \psi \wedge \phi \vdash \theta}$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \psi \vee \phi} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \psi \vee \phi} \qquad \frac{\Gamma, \psi \vdash \theta \qquad \Gamma, \phi \vdash \theta}{\Gamma, \psi \vee \phi \vdash \theta}$$

$$\frac{\Gamma \vdash \psi \qquad \Gamma, \phi \vdash \theta}{\Gamma, \psi \to \phi \vdash \theta} \qquad \frac{\Gamma, \psi \vdash \phi}{\Gamma \vdash \psi \to \phi}$$

Local Proof System (cont.)

$$\frac{\Gamma, \phi \vdash \bot}{\Gamma \vdash \neg\phi} \qquad \frac{}{\Gamma, \bot \vdash \phi}$$

$$\frac{\Gamma_1, \phi, \psi, \Gamma_2 \vdash \theta}{\Gamma_1, \psi, \phi, \Gamma_2 \vdash \theta} \qquad \frac{\Gamma, \psi, \psi \vdash \phi}{\Gamma, \psi \vdash \phi}$$

Local Proof System: Soundness

The local proof system is **sound**.

Proof

First, recall that for any Γ and θ:

$$\Gamma \vdash \theta \text{ is valid iff. } \mathrm{Pr}_\rho \left([\![(\wedge\Gamma) \wedge \neg\phi]\!]_{\mathbb{M}}^{\eta,\rho} \right) \text{ is negligible.} \qquad (\dagger)$$

Local Proof System: Soundness

We will only detail one rule, say:

$$\frac{\Gamma, \psi \vdash \theta \qquad \Gamma, \phi \vdash \theta}{\Gamma, \psi \vee \phi \vdash \theta.}$$

By the previous remark (†), since $(\Gamma, \psi \vdash \theta)$ and $(\Gamma, \phi \vdash \theta)$ are valid

- $\mathsf{Pr}_\rho\left([\![(\wedge\Gamma) \wedge \psi \wedge \neg\theta]\!]_{\mathbb{M}}^{\eta,\rho}\right)$ is negligible.
- $\mathsf{Pr}_\rho\left([\![(\wedge\Gamma) \wedge \phi \wedge \neg\theta]\!]_{\mathbb{M}}^{\eta,\rho}\right)$ is negligible.

Since the union of two negligible (η-indexed families of) events is a negligible (η-indexed families of) events,

$$\begin{aligned} & \mathsf{Pr}_\rho\left([\![\big((\wedge\Gamma) \wedge \psi \wedge \neg\theta\big) \vee \big((\wedge\Gamma) \wedge \phi \wedge \neg\theta\big)]\!]_{\mathbb{M}}^{\eta,\rho}\right) \text{ is negligible} \\ \Leftrightarrow\ & \mathsf{Pr}_\rho\left([\![(\wedge\Gamma) \wedge (\psi \vee \phi) \wedge \neg\theta]\!]_{\mathbb{M}}^{\eta,\rho}\right) \text{ is negligible} \end{aligned}$$

Hence using (†) again, $\Gamma, \psi \vee \phi \vdash \theta$ is valid.

Authentication Protocols

Cryptographic Rule: Collision Resistance

Cryptographic Hash

A **keyed cryptographic hash** $\mathsf{H}(\_,\_)$ is **computationally collision resistant** if no PPTM adversary can built collisions, even when it has access to a hashing **oracle**.

More precisely, a hash is *collision resistant under hidden key attacks* (CR-HK) iff for every PPTM $\mathcal{A}$, the following quantity:

$$\Pr_{\mathsf{k}}\left(\mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = \langle m_1\,,\, m_2\rangle, m_1 \neq m_2 \text{ and } \mathsf{H}(m_1,\mathsf{k}) = \mathsf{H}(m_2,\mathsf{k})\right)$$

is negligible, where k is drawn uniformly in $\{0,1\}^\eta$.

CR Rule

Collision Resistance

If H is a CR-HK function, then the *ground* rule:

$$\frac{}{\mathsf{H}(m_1, \mathsf{k}) = \mathsf{H}(m_2, \mathsf{k}) \vdash m_1 = m_2}\ \mathrm{CR}$$

is sound, when k appears only in H key positions in m_1, m_2.

Authentication Protocols

Cryptographic Rule: Message Authentication Code

Message Authentication Code

A **message authentication code** is a symmetric cryptographic schema which:

- create **message authentication codes** using $\text{mac}_{\cdot}(\cdot)$
- **verifies** mac using $\text{verify}_{\cdot}(\cdot, \cdot)$

It must satisfies the functional equality:

$$\text{verify}_{k}(\text{mac}_{k}(m), m) = \text{true}$$

MAC Security

A MAC must be **computationally unforgeable**, even when the adversary has access to a mac and verify **oracles**.

A MAC is *unforgeable against chosen-message attacks* (EUF-CMA) iff for every PPTM $\mathcal{A}$, the following quantity:

$$\Pr_{k}\left(\begin{array}{r}\mathcal{A}^{\mathcal{O}_{\mathsf{mac}_k(\cdot)},\mathcal{O}_{\mathsf{verify}_k(\cdot,\cdot)}}(1^\eta) = \langle m\,,\,\sigma\rangle,\ m \text{ not queried to } \mathcal{O}_{\mathsf{mac}_k(\cdot)}\\ \text{and } \mathsf{verify}_k(\sigma, m) = 1\end{array}\right)$$

is negligible, where k is drawn uniformly in $\{0,1\}^\eta$.

EUF-MAC Rule

Take two messages s, m and a key $\mathsf{k} \in \mathcal{N}$ such that

- s and m are ground.
- $\mathsf{k} \in \mathcal{N}$ appears only in mac or verify key positions in s, m.

Key Idea

To build a rule for EUF-CMA, we proceed as follow:

- Compute $[\![s, m]\!]$ bottum-up, calling $\mathcal{O}_{\mathsf{mac}_{\mathsf{k}}(\cdot)}$ and $\mathcal{O}_{\mathsf{verify}_{\mathsf{k}}(\cdot,\cdot)}$ if necessary.
- Log all sub-terms $\mathbb{S}_{\mathsf{mac}}(s, m)$ sent to $\mathcal{O}_{\mathsf{mac}_{\mathsf{k}}(\cdot)}$.

$\Rightarrow$ If $\mathsf{verify}_{\mathsf{k}}(s, m)$ then $m = u$ for some $u \in \mathbb{S}_{\mathsf{mac}}(s, m)$.

💡 *$\mathbb{S}_{mac}(s, m)$ are the **calls** to $\mathcal{O}_{mac_k(\cdot)}$ needed to compute s, m.*

EUF-MAC Rule

$\mathbb{S}_{\mathsf{mac}}(\cdot)$ defined by induction on ground terms:

$$\mathbb{S}_{\mathsf{mac}}(\mathsf{n}) \stackrel{\text{def}}{=} \emptyset$$

$$\mathbb{S}_{\mathsf{mac}}(\mathsf{verify}_{\mathsf{k}}(u_1, u_2)) \stackrel{\text{def}}{=} \mathbb{S}_{\mathsf{mac}}(u_1) \cup \mathbb{S}_{\mathsf{mac}}(u_2)$$

$$\mathbb{S}_{\mathsf{mac}}(\mathsf{mac}_{\mathsf{k}}(u)) \stackrel{\text{def}}{=} \{u\} \cup \mathbb{S}_{\mathsf{mac}}(u)$$

$$\mathbb{S}_{\mathsf{mac}}(f(u_1, \ldots, u_n)) \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq n} \mathbb{S}_{\mathsf{mac}}(u_i) \qquad \text{(for other cases)}$$

EUF-MAC Rule

Message Authentication Code Unforgeability

If mac is an EUF-CMA function, then the *ground* rule:

$$\frac{}{\mathsf{verify}_{\mathsf{k}}(s, m) \vdash \bigvee_{u \in \mathcal{S}} m = u} \text{ EUF-MAC}$$

is sound, when:

- $\mathcal{S} = \mathbb{S}_{\mathsf{mac}}(s, m)$;
- $\mathsf{k} \in \mathcal{N}$ appears only in mac or verify key positions in s, m.

Example

If t_1 t_2 and t_3 are terms which do not contain k, then:

$$\Phi \equiv \mathsf{mac}_{\mathsf{k}}(t_1), \mathsf{mac}_{\mathsf{k}}(t_2), \mathsf{mac}_{\mathsf{k_0}}(t_3)$$

$$\left[\mathsf{verify}_{\mathsf{k}}(g(\Phi), \mathsf{n}) \rightarrow (\mathsf{n} = t_1 \vee \mathsf{n} = t_2)\right]$$

EUF-MAC Rule: Exercise

Exercise

Assume mac is EUF-CMA. Show that the following rule is sound:

$$\frac{}{\mathsf{verify}_{\mathsf{k}}(\text{if } b \text{ then } s_0 \text{ else } s_1, m) \vdash \bigvee_{u \in \mathcal{S}_1 \cup \mathcal{S}_2} m = u}$$

when b, s_0, s_1, m are *ground* terms, and:

- $\mathcal{S}_i = \{u \mid \mathsf{mac}_{\mathsf{k}}(u) \in \mathbb{S}_{\mathsf{mac}}(s_i, m)\}$, for $i \in \{0, 1\}$;
- k appears only in mac or verify key positions in s_0, s_1, m.

Remark: we do not make *any* assumption on b, except that it is ground. E.g., we can have $b \equiv (\mathbf{att}(\mathsf{k}) = \mathsf{mac}_{\mathsf{k}}(0))$.

Authentication Protocols

Authentication of the Hash-Lock Protocol

Authentication: Hash-Lock

Theorem

Assuming that the hash function is EUF-CMA[2], the Hash-Lock protocol provides **authentication**, i.e. for any identity $a \in \mathcal{I}$, for any $\mathtt{tr} \in \mathcal{T}_{io}$, $\mathtt{tr}_1 : \mathtt{R}_\mathtt{j}^1$ and $\mathtt{tr}_3 : \mathtt{R}_\mathtt{j}^2$ s.t.:

$$\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr}$$

the following formula is valid:

$$\mathsf{accept}^{\mathsf{A}} @ \mathtt{tr}_3 \vdash \bigvee_{\substack{\mathtt{tr}_2 : \mathtt{A}_\mathtt{i} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}} \begin{array}{l} \mathsf{out} @ \mathtt{tr}_1 = \mathsf{in} @ \mathtt{tr}_2 \wedge \\ \mathsf{out} @ \mathtt{tr}_2 = \mathsf{in} @ \mathtt{tr}_3 \end{array}$$

[2] Taking $\mathsf{verify}_{\mathsf{k}}(s, m) \stackrel{\text{def}}{=} s = \mathsf{H}(m, \mathsf{k})$.

Authentication: Hash-Lock

Proof. Let $\mathsf{a} \in \mathcal{I}$, and let $\mathtt{tr} \in \mathcal{T}_{\mathsf{io}}$, $\mathtt{tr}_1 : \mathtt{R}^1_\mathtt{j}$ and $\mathtt{tr}_3 : \mathtt{R}^2_\mathtt{j}$ be s.t.:

$$\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr}$$

We let:

$$t_{\mathsf{conc}} \stackrel{\mathsf{def}}{=} \bigvee_{\substack{\mathtt{tr}_2 : \mathtt{A}_\mathtt{i} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}} \mathsf{out}@\mathtt{tr}_1 = \mathsf{in}@\mathtt{tr}_2 \wedge \mathsf{out}@\mathtt{tr}_2 = \mathsf{in}@\mathtt{tr}_3$$

We must prove that the following local judgement is valid:

$$\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3 \vdash t_{\mathsf{conc}}$$

i.e. that:

$$\pi_2(\mathsf{in}@\mathtt{tr}_3) = \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathsf{j}} \,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle, \mathsf{k}_\mathsf{A}) \vdash t_{\mathsf{conc}}$$

Authentication: Hash-Lock

We use the EUF-MAC rule on the equality:

$$\pi_2(\mathsf{in}@\mathtt{tr}_3) = \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}} , \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle, \mathsf{k}_\mathsf{A}) \qquad (\dagger)$$

The terms above are ground, and the key k_A is correctly used in them.
Moreover, the set of *honest* hashes using key k_A appearing in (†), excluding the top-level hash, is:

$$\begin{aligned} & \mathbb{S}_{\mathsf{mac}}(\pi_2(\mathsf{in}@\mathtt{tr}_3), \langle \mathsf{n}_{\mathsf{R},\mathtt{j}} , \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle) \\ = {} & \mathbb{S}_{\mathsf{mac}}(\mathsf{in}@\mathtt{tr}_3) \\ = {} & \{\mathsf{H}(\langle \mathsf{in}@\mathtt{tr}_2 , \mathsf{n}_{\mathsf{A},\mathtt{i}}\rangle, \mathsf{k}_\mathsf{A}) \mid \mathtt{tr}_2 : \mathtt{A}_\mathtt{i} < \mathtt{tr}_3\} \end{aligned}$$

The hashes in the reader's outputs can be seen as verify checks, and can therefore be ignored.

Authentication: Hash-Lock

Hence using EUF-MAC plus some basic reasoning, we have:

$$\dfrac{\dfrac{\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3, \begin{array}{l}\langle \mathsf{in}@\mathtt{tr}_2\,,\, \mathsf{n}_{\mathsf{A},\mathtt{i}}\rangle = \\ \langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle\end{array} \vdash t_{\mathsf{conc}} \qquad \text{for every } \mathtt{tr}_2 : \mathtt{A}_\mathtt{i} < \mathtt{tr}_3}{\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3, \bigvee_{\mathtt{tr}_2 : \mathtt{A}_\mathtt{i} < \mathtt{tr}_3} \begin{array}{l}\langle \mathsf{in}@\mathtt{tr}_2\,,\, \mathsf{n}_{\mathsf{A},\mathtt{i}}\rangle = \\ \langle \mathsf{n}_{\mathsf{R},\mathtt{j}}\,,\, \pi_1(\mathsf{in}@\mathtt{tr}_3)\rangle\end{array} \vdash t_{\mathsf{conc}}}}{\mathsf{accept}^{\mathsf{A}}@\mathtt{tr}_3 \vdash t_{\mathsf{conc}}}$$

Authentication: Hash-Lock

Assuming that the pair and projections satisfy:

$$\overline{[\pi_1\langle x\,,\, y\rangle = x]} \qquad\qquad \overline{[\pi_2\langle x\,,\, y\rangle = y]}$$

We only have to show that for every $\mathtt{tr}_2 : \mathtt{A}_\mathtt{i} < \mathtt{tr}_3$:

$$\Gamma \vdash t_{\mathsf{conc}}$$

is valid, where:

$$\Gamma \stackrel{\mathsf{def}}{=} \left(\mathsf{accept}^{\mathsf{A}} @\mathtt{tr}_3,\ \mathsf{in}@\mathtt{tr}_2 = \mathsf{n}_{\mathsf{R},\mathtt{j}},\ \mathsf{n}_{\mathsf{A},\mathtt{i}} = \pi_1(\mathsf{in}@\mathtt{tr}_3)\right)$$

Authentication: Hash-Lock

Since $\mathtt{tr}_1 : \mathtt{R}^1_\mathtt{j} < \mathtt{tr}_3$ we know that:

$$\mathtt{out}@\mathtt{tr}_1 \stackrel{\mathsf{def}}{=} \mathsf{n}_{\mathsf{R},\mathtt{j}}$$

Moreover:

$$\mathtt{out}@\mathtt{tr}_2 \stackrel{\mathsf{def}}{=} \langle \mathsf{n}_{\mathsf{A},\mathtt{i}} \, , \, \mathsf{H}(\langle \mathsf{in}@\mathtt{tr}_2 \, , \, \mathsf{n}_{\mathsf{A},\mathtt{i}} \rangle, \mathsf{k}_\mathsf{A}) \rangle$$

Hence:

$$\Gamma \vdash \pi_1(\mathtt{out}@\mathtt{tr}_2) = \pi_1(\mathsf{in}@\mathtt{tr}_3) \qquad (\diamond)$$

Similarly:

$$\begin{aligned} \Gamma \vdash \pi_2(\mathtt{out}@\mathtt{tr}_2) &= \mathsf{H}(\langle \mathsf{in}@\mathtt{tr}_2 \, , \, \mathsf{n}_{\mathsf{A},\mathtt{i}} \rangle, \mathsf{k}_\mathsf{A}) \\ &= \mathsf{H}(\langle \mathsf{n}_{\mathsf{R},\mathtt{j}} \, , \, \pi_1(\mathsf{in}@\mathtt{tr}_3) \rangle, \mathsf{k}_\mathsf{A}) \\ &= \pi_2(\mathsf{in}@\mathtt{tr}_3) \end{aligned}$$

Consequently:

$$\Gamma \vdash \pi_2(\mathtt{out}@\mathtt{tr}_2) = \pi_2(\mathsf{in}@\mathtt{tr}_3) \qquad (\star)$$

Authentication: Hash-Lock

Assuming that the pair and projections satisfy the property:

$$\overline{\left[(\pi_1\ x = \pi_1\ y) \rightarrow (\pi_2\ x = \pi_2\ y) \rightarrow x = y\right]}$$

We deduce from ($\star$) and ($\diamond$) that:

$$\Gamma \vdash \mathsf{out}@\mathtt{tr}_2 = \mathsf{in}@\mathtt{tr}_3$$

Putting everything together, we get:

$$\Gamma \vdash \mathsf{out}@\mathtt{tr}_1 = \mathsf{in}@\mathtt{tr}_2 \wedge \mathsf{out}@\mathtt{tr}_2 = \mathsf{in}@\mathtt{tr}_3 \qquad (\ddagger)$$

Authentication: Hash-Lock

Recall that:

$$t_{\text{conc}} \stackrel{\text{def}}{=} \bigvee_{\substack{\texttt{tr}_2:\texttt{A}_\texttt{i} \\ \texttt{tr}_1 \leq \texttt{tr}_2 \leq \texttt{tr}_3}} \text{out}@\texttt{tr}_1 = \text{in}@\texttt{tr}_2 \wedge \text{out}@\texttt{tr}_2 = \text{in}@\texttt{tr}_3$$

and we must show that $\Gamma \vdash t_{\text{conc}}$. Hence, using (‡), it only remains to prove that whenever $\texttt{tr}_2 < \texttt{tr}_1$, we have:

$$\Gamma,\ \text{out}@\texttt{tr}_1 = \text{in}@\texttt{tr}_2,\ \text{out}@\texttt{tr}_2 = \text{in}@\texttt{tr}_3 \vdash \bot$$

This follows from the independence rule:

$$\frac{}{[t \neq \mathsf{n}]} \; {=}\text{-IND} \qquad \text{when } t \text{ is ground and } \mathsf{n} \notin \text{st}(t)$$

using the fact that:

$$\text{out}@\texttt{tr}_1 \stackrel{\text{def}}{=} \mathsf{n}_{\mathsf{R},\mathsf{j}}$$

and that if $\texttt{tr}_2 < \texttt{tr}_1$ then $\mathsf{n}_{\mathsf{R},\mathsf{j}} \notin \text{st}(\text{in}@\texttt{tr}_2)$.

Authentication Protocols

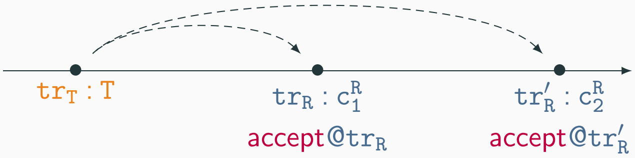
Beyond Authentication

Beyond Authentication

Authentication, which states that we must have:

$\forall \mathtt{tr}_\mathtt{R} : \mathtt{R}.\ \exists \mathtt{tr}_\mathtt{T} : \mathtt{T}.$

does not exclude the scenario:



Replay Attack

This is a **replay attack**: the **same message** (or partial transcript), when replayed, is **accepted again** by the server.

This can yield real-word **attacks**. E.g. an adversary can open a door at will once it eavesdropped one honest interaction.

Example

The following protocol, called Basic Hash, suffer from such attacks:

$$
\begin{aligned}
&\mathsf{T}(\mathsf{A}, \mathtt{i}) : \nu\, \mathsf{n}_{\mathsf{A},\mathtt{i}}.\ \mathbf{out}(\mathtt{A}_\mathtt{i}, \langle \mathsf{n}_{\mathsf{A},\mathtt{i}}\,, \mathsf{H}(\mathsf{n}_{\mathsf{A},\mathtt{i}}, \mathsf{k}_\mathsf{A})\rangle)\\
&\mathsf{R}(\mathtt{j}) \quad\ : \mathbf{in}(\mathtt{R}^2_\mathtt{j}, \mathsf{y}).\ \mathbf{out}(\mathtt{R}^2_\mathtt{j}, \text{if } \bigvee_{\mathsf{A}\in\mathcal{I}} \pi_2(\mathsf{y}) = \mathsf{H}(\pi_1(\mathsf{y}), \mathsf{k}_\mathsf{A}))\\
&\qquad\qquad\qquad\qquad\qquad\quad \text{then ok}\\
&\qquad\qquad\qquad\qquad\qquad\quad \text{else ko}
\end{aligned}
$$

Injective Authentication

The **authentication** property is too *weak* for many real-world application.

To prevent replay attacks, we require that the protocol provides a **stronger** property, **injective authentication**.

Injective Authentication: Hash-Lock

The following formulas encode the fact that the **Hash-Lock** protocol provides **injective authentication**:

$\forall \mathsf{A} \in \mathcal{I}.\ \forall \mathtt{tr} \in \mathcal{T}_{io}.\ \forall \mathtt{tr}_1 : \mathtt{R}^1_{\mathtt{j}}, \mathtt{tr}_3 : \mathtt{R}^2_{\mathtt{j}}$ s.t. $\mathtt{tr}_1 < \mathtt{tr}_3 \leq \mathtt{tr}$

$$\mathsf{accept}^{\mathsf{A}} @ \mathtt{tr}_3 \rightarrow \bigvee_{\substack{\mathtt{tr}_2 : \mathtt{A}_\mathtt{i} \\ \mathtt{tr}_1 \leq \mathtt{tr}_2 \leq \mathtt{tr}_3}} \begin{array}{l} \mathsf{out} @ \mathtt{tr}_1 = \mathsf{in} @ \mathtt{tr}_2 \wedge \\ \mathsf{out} @ \mathtt{tr}_2 = \mathsf{in} @ \mathtt{tr}_3 \end{array}$$

$$\wedge \bigwedge_{\substack{\mathtt{tr}'_1 : \mathtt{R}^1_\mathtt{k},\ \mathtt{tr}'_3 : \mathtt{R}^2_\mathtt{k} \\ \mathtt{tr}'_1 < \mathtt{tr}'_3 \leq \mathtt{tr}}} \left(\begin{array}{l} \mathsf{accept}^{\mathsf{A}} @ \mathtt{tr}'_3 \wedge \\ \mathsf{out} @ \mathtt{tr}_2 = \mathsf{in} @ \mathtt{tr}'_3 \end{array} \rightarrow j = k \right)$$

[1] D. Baelde, S. Delaune, and L. Hirschi.
Partial order reduction for security protocols.
In *CONCUR*, volume 42 of *LIPIcs*, pages 497–510. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.